Please Amend the Specification as Follows:

1. Amend the first two paragraphs of the Specification as follows:

Serial number ~~xx/xxx,xxx~~ 10/600,880 entitled "System and Method for Ensuring Memory Coherency within a Multiprocessor Shared Memory System that Provides Data Ownership Prior to Invalidation", Attorney Docket Number RA-5614, filed on even date herewith, and incorporated herein by reference in its entirety.

Serial number ~~xx/xxx,xxx~~ 10/600,218 entitled "Data Acceleration Mechanism for a Multiprocessor Shared Memory System", Attorney Docket Number RA-5470, filed on even date herewith, and incorporated herein by reference in its entirety.

2. Amend the first paragraph on page 4 as follows:

A system and method are provided for tracking memory requests within a data processing system. The system includes a request tracking circuit that is coupled to receive requests for data from multiple processors. After a request is received and before it is forwarded to the memory for processing, a record is created within the request tracking circuit that stores request information. For example, this information may identify the request address, the processor that issued the request, as well as the request type. The request tracking circuit then determines whether any other requests are pending for the same memory address. If not, the request is forwarded to the memory. Otherwise, a request is not issued to memory, and instead, the newly-created record is ~~instead~~ associated with one or more other records tracking requests to the same address. In one embodiment, this association is created by forming a linked list of records. These records may be linked in an order that indicates the time-order in which the respective requests were received.

3. Amend the second full paragraph on page 14 as follows:

The logic of Figure 2 may best be understood by considering the following example. Assume that IP 110A is requesting ownership of a cache line for update purposes. A cache miss results in SLC 108A, and a request is therefore issued on processor bus 104A to request the data. This request will cause ~~in~~ any other SLC

on processor bus 104A to return any updated data copy to SLC 108A on the processor bus. If the request is for data ownership, it will also result in invalidation of any other copies retained by the SLCs of processing node 120A.

4. Amend the second and third full paragraphs of page 17 as follows:

After SCD receives the request for the cache line, directory 101 is referenced to determine whether any of the one or more other processing nodes within the system stores a copy of the requested data. In the current example, it will be assumed the most recent copy of the requested data is available within ~~SND~~ SCD 100. This data is provided to PND 102A along with the original transaction identifier and a response type of ownership-with-data. This response type indicates that there is no outstanding response associated with the data. Other cases involving the return of data while some responses are still outstanding are discussed below.

When the PND receives the data and response from ~~SND~~ SCD 100, the transaction identifier provided with the response is used by LT control logic 203 to reference LT 212 and retrieve the deferred identifier for this request from DID field 238. The returned data is routed from input queue 240 to output queue 242, and is provided on processor bus 104A. In one embodiment, this data is provided to SLC 108A during what is known as a "deferred phase". A deferred phase is one of the ways a PND 102 provides data following the issuance of a deferred response. During a deferred phase, PND 102 places an encoded value on processor bus 104A indicating that a deferred phase is occurring, along with the deferred identifier retrieved from LT 212. The deferred identifier is used by the target SLC to match the returned data with the original cache line request. In the current example, after SLC 108A receives and processes the deferred phase, the data will be forwarded to IP 110A to satisfy the initial request.

5. Amended the second full paragraph on page 18 as follows:

According to one aspect of the invention, the lookup table used to control the unlinking of LT entries may be programmable, and may be stored within a memory

such as LT control store 288. This lookup table could be modified using a scan-set interface, as is known in the art. By allowing the information to be programmable, the type of actions taken to unlink the entries within LT can change as the needs of the system change. For example, if different types of processors are coupled to processor bus 104A, the types of requests that may be issued to obtain data in various situations may change. This can be accomplished merely by ~~modified~~ modifying the control store. As a general rule, normal processing activities must be halted before modifying LT control store 288 so as to avoid the occurrence of errors.

6. Amend the last full paragraph on page 21 as follows:

Although the replacement operation is not performed in the current instance, the LT control logic 203 will process the original request entry associated with SLC 108A in the manner discussed above. That is, after the original request entry is removed from LT 212, any LT entries linked to this entry are likewise unlinked and removed. As each entry is unlinked, a request for data is issued to processor bus 104A, followed by a deferred reply that is autonomously associated with the request for data. During this process, a request for data made to processor bus 104A may result in a miss if that data was returned to PND 102A during a write back operation. ~~In~~ If this occurs, the subsequent deferred response will provide a copy of the data retrieved from cache 206 to whichever SLC 108 is associated with the next request in the linked list.